

Embedded boundary grid generation using the divergence theorem, implicit functions, and constructive solid geometry

Terry J Ligocki, Peter O Schwartz, Julie Percelay and Phillip Colella

Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

E-mail: TJLigocki@lbl.gov

Abstract.

To construct finite-volume methods for PDEs in arbitrary dimension to arbitrary accuracy in the presence of irregular boundaries, we show that estimates of moments, integrals of monomials, over various regions are all that are needed. If implicit functions are used to represent the irregular boundary, the needed moments can be computed straightforwardly and robustly by using the divergence theorem, Taylor expansions, least squares, recursion, and 1D root finding. Neither a geometric representation of the irregular boundary nor its interior is ever needed or computed. The implicit function representation is general and robust. Implicit functions can be combined via constructive solid geometry to form complex boundaries from a rich set of primitives including interpolants of sampled data, for example, 2D/3D image data and digital elevation maps.

1. Finite-volume methods and flux computations

The overall goal of this work is to numerically compute solutions to PDE on domains with irregular boundaries in an arbitrary dimension Cartesian space, \mathbb{R}^D , to an arbitrary accuracy by using finite-volume methods [1]. The PDE are written as conservation laws that contain a divergence of a “flux”, $\nabla \cdot \mathbf{F}$, for example,

$$\nabla \cdot \beta(\nabla u) = \rho, \quad \mathbf{F}(u) = \beta(\nabla u) \quad (\text{Elliptic PDE})$$

$$\frac{\partial u}{\partial t} = \nabla \cdot \rho(\nabla u), \quad \mathbf{F}(u) = \rho(\nabla u) \quad (\text{Parabolic PDE})$$

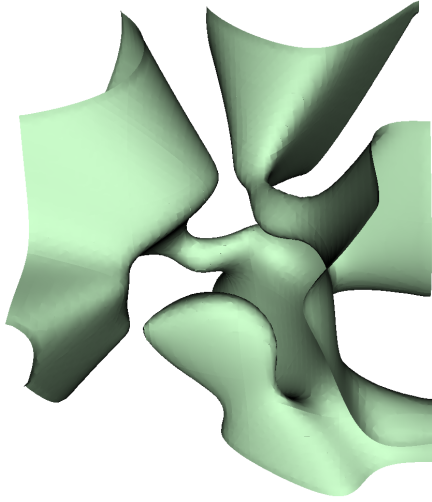
$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u) = S(u), \quad \mathbf{F}(u) \text{ is given.} \quad (\text{Hyperbolic PDE})$$

The PDE are solved on a discretization of an irregular domain, Ω , defined implicitly using an “implicit” function, $\phi : \mathbb{R}^D \rightarrow \mathbb{R}$, namely,

$$\Omega = \{\mathbf{x} : \mathbf{x} \in \mathbb{R}^D, \phi(\mathbf{x}) < 0\}. \quad (1)$$

Implicit functions are chosen to represent Ω for several reasons:

- The implicit function, $\phi(\mathbf{x})$, is defined everywhere in \mathbb{R}^D , and thus Taylor expansions of ϕ and functions of ϕ (e.g., $\mathbf{n}(\mathbf{x}) = \nabla \phi(\mathbf{x}) / |\nabla \phi(\mathbf{x})|$) can be computed anywhere they exist if ϕ is smooth enough.



(a) Polynomial function of three variables



(b) SF Bay digital elevation map data

Figure 1. Domains generated via an implicit functions.

- Implicit functions can be used to represent a rich set of geometric shapes (see figure 1) either in functional form or as interpolants of discrete, sampled data (e.g., 2D/3D image data [3, 4, 5], digital elevation maps).
- Implicit functions can be easily restricted to lower dimensions. This is exploited in the algorithm that computes moments (see Section 2).
- Implicit functions can be extended to arbitrary dimensions allowing computations to be done in phase spaces or space-time.
- Ω can implicitly evolve in time if ϕ is allowed to change with time, possibly as a function of other PDE variables or time directly.

In this embedded boundary approach, Ω will be discretized on a set of control volumes, \mathcal{V} , formed by intersecting rectangular cells with Ω :

$$\mathcal{V} = \{[\mathbf{i}h, (\mathbf{i} + \mathbf{e})h] \cap \Omega : \mathbf{i} \in \mathbb{Z}^D\}, \quad (2)$$

where $\mathbf{e} \in \mathbb{Z}^D$ and all its components are one and h is the size of the rectangular cells that are, in this case, square.

Given this formulation, the integral of the divergence of the vector field, $\nabla \cdot \mathbf{F}$, over the control volume, $V \in \mathcal{V}$, can be transformed by using the divergence theorem into an integral over the boundary of the volume, $A = \partial V$:

$$\int_V \nabla \cdot \mathbf{F} dV = \int_A \mathbf{F} \cdot \mathbf{n} dA \quad (3)$$

where \mathbf{n} is the normal to A pointing out of V . In control volumes that do not intersect the boundary of Ω , this can be rewritten as A_-^d : $\mathbf{F}_- \cdot \mathbf{n} = -F_-^d$ on the low faces and as A_+^d : $\mathbf{F}_+ \cdot \mathbf{n} = F_+^d$ on the high faces by noting that on the faces the normal points in a coordinate direction, d . The result is the canonical “(sum of) difference in fluxes (integrated over faces)” formulation of finite volume methods. These methods are as accurate as the approximations to integrals of F_\pm^d over the faces.

With these definitions, a finite-volume method for control volumes containing embedded boundaries can be defined by using the divergence theorem:

$$\int_V \nabla \cdot \mathbf{F} dV = \sum_{\pm = +, -} \sum_{d=1}^D \pm \int_{A_{\pm}^d} F_{\pm}^d dA + \int_{A_{EB}} \mathbf{F} \cdot \mathbf{n}_{EB} dA, \quad (4)$$

where $A_{\pm}^d = \{x : x \in V, x_d = (i_d + \frac{1}{2} \pm \frac{1}{2})h\}$, $A_{EB} = V \cap \partial\Omega$, and $\mathbf{n}_{EB} = \nabla\phi/|\nabla\phi|$. Note: \mathbf{n}_{EB} is defined *everywhere* in Ω and not simply on $\partial\Omega$. This equation has still not been discretized; that is, it is exact. The integrals over the boundary are discretized by using an P^{th} order Taylor expansion of \mathbf{F} and F_{\pm}^d about points $\mathbf{x}_{EB} \in A_{EB}$ and $\mathbf{x}_{\pm}^d \in A_{\pm}^d$ that approximate the centroids of the domains of integration. In addition, we can expand \mathbf{n}_{EB} about some point in the cell (which we take to be the origin of our coordinates), since it is a known smooth function. Using these Taylor expansions, we can rewrite equation (4) as follows:

$$\begin{aligned} \int_V \nabla \cdot \mathbf{F} dV = & \sum_{0 \leq |\mathbf{p}| \leq P} \frac{1}{\mathbf{p}!} \left(\sum_{\pm = +, -} \sum_{d=1}^D \pm (\nabla^{\mathbf{p}} F_{\pm}^d) \int_{A_{\pm}^d} (\mathbf{x} - \mathbf{x}_{\pm}^d)^{\mathbf{r}} dA \right. \\ & \left. + \nabla^{\mathbf{p}} \mathbf{F} \cdot \sum_{0 \leq |\mathbf{r}| \leq R} \frac{\nabla^{\mathbf{r}} \mathbf{n}_{EB}}{\mathbf{r}!} \int_{A_{EB}} (\mathbf{x} - \mathbf{x}_{EB})^{\mathbf{p}} \mathbf{x}^{\mathbf{r}} dA \right) + O(h^{D+P+R}). \end{aligned} \quad (5)$$

Assuming \mathbf{F} is discretized on the rectangular grid covered by Ω , one can approximate $\nabla^{\mathbf{p}} F_{\pm}^d$ and $\nabla^{\mathbf{r}} \mathbf{F}$ to the appropriate order by finite differences. The remaining information required is in the form of moments, namely, integrals of polynomials computed over the V , A_{\pm}^d , and A_{EB} . Note that \mathbf{x}_{\pm}^d and \mathbf{x}_{EB} can also be computed from moments. Computing these is the analogue of grid generation for the embedded boundary method. No explicit representation of the irregular boundary is ever needed.

2. Moment computations

Approximating the needed moments to the necessary order can be accomplished by taking all monomials, $\mathbf{x}^{\mathbf{p}}$, of a given degree, P , where \mathbf{p} is a multi-index and $|\mathbf{p}| = P$, and substituting $\mathbf{F}(\mathbf{x}) = \mathbf{x}^{\mathbf{p}} \mathbf{e}_d$ into equation (4), where \mathbf{e}_d is the unit vector in direction d . Note that $\partial \mathbf{x}^{\mathbf{p}} / \partial x_d = p_d \mathbf{x}^{\mathbf{p} - \mathbf{e}_d}$ and replacing \mathbf{n}_{EB} by its Taylor expansion; then each monomial yields the set of D equations defined by

$$\begin{aligned} p_d \int_V \mathbf{x}^{\mathbf{p} - \mathbf{e}_d} dV - n_d \int_{A_{EB}} \mathbf{x}^{\mathbf{p}} dA = & \sum_{\pm = +, -} \pm \int_{A_{\pm}^d} \mathbf{x}^{\mathbf{p}} dA \\ & + \sum_{1 \leq |\mathbf{r}| \leq R} \left(\frac{\nabla^{\mathbf{r}} n_d}{\mathbf{r}!} \int_{A_{EB}} \mathbf{x}^{(\mathbf{p} + \mathbf{r})} dA \right) + O(h^{D+P+R}) \end{aligned} \quad (6)$$

If the integrals on the LHS are treated as unknowns and the RHS is treated as known, then the set of equations formed in this way for all monomials $\mathbf{x}^{\mathbf{p}}$ where $|\mathbf{p}| = P$ forms a system of linear equations. The number of unknowns is $N_{P-1} + N_P$ (where N_P is the number of monomials of degree P) and number of equations is DN_P . Since $N_{P-1} < N_P$, if $D \geq 2$, this is an overdetermined system and can be solved by least squares. The one-dimensional case can be solved explicitly by using 1D root-finding techniques and integration of monomials (see below).

Examining the RHS shows that the first integral is of the same form as the volume integrals on the LHS except the integral has been restricted to one less dimension and the monomial degree

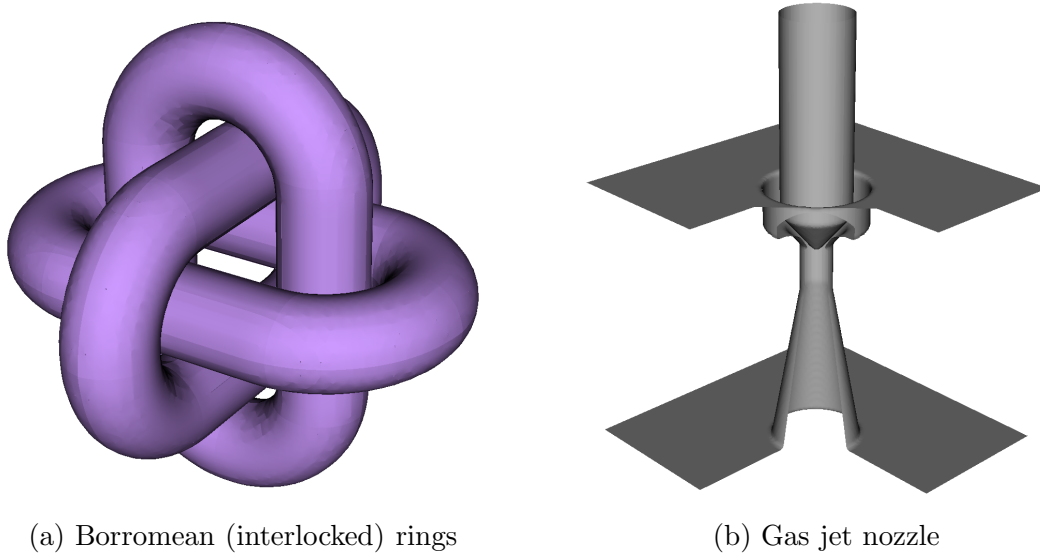


Figure 2. Domains generated via constructive solid geometry and multiple implicit functions.

is one more. Hence, the order of accuracy is the same, $((P + 1) + (D - 1) + R = P + D + R)$. Using recursion, we can reduce the dimension until $D = 1$. At this point all the moments can be computed over A_d because A_d is the intersection of V with a line parallel to the x_d axis and the monomial only has one variable that isn't constant. The intersection can be found by using robust 1D root finding algorithms to get the end points of the 1D integral, which is then computed explicitly.

The second integral on the RHS involves moments over the A_{EB} in the current dimension but the degree has increased to $P + |r|$. Thus, they can be computed recursively. The recursion terminates because as the degree of the monomial moments increases, the dimension and order of accuracy stays the same. Thus, the value of R (the number of terms in the Taylor series) in the recursion decreases by $|r|$. Eventually, there are no terms of the Taylor series on the RHS so this recursion terminates.

Overall, the algorithm proceeds by setting up the equations to compute the needed moments over the needed regions as a least squares problem. Then the RHS is recursively computed recursing in both dimension and monomial degree. Finally, the original least squares problems are solved for the unknown moments, and these are used in the flux computations. Since the least squares problems are always well conditioned and the only computations involving the Ω are the intersections of the Ω with lines parallel to the coordinate axes, this algorithm is very robust and straightforward to implement.

3. Complex irregular boundaries

In order to construct more complex irregular boundaries, implicit functions can be composed into more complex implicit functions using constructive solid geometry, CSG. To do so, one must define the complement, intersection, and union of irregular domains, Ω_i , defined by implicit functions, ϕ_i . To this end we use the following correspondences.

$$\begin{aligned}
 \Omega_i^{\text{complement}} &\Leftrightarrow -\phi_i \\
 \Omega^{\text{intersection}} &\Leftrightarrow \max_i \phi_i \\
 \Omega^{\text{union}} &\Leftrightarrow \min_i \phi_i
 \end{aligned}$$

Further, coordinate transformations, ψ , of Ω can be implemented as

$$\Omega_\psi = \{\mathbf{x} : \mathbf{x} \in \mathbb{R}^D, \phi(\psi^{-1}(\mathbf{x})) < 0\}.$$

Examples of ψ include rotations, translations, and scaling. In addition to being straightforward to implement by using implicit functions, CSG is a very robust and intuitive method for building complex geometry objects and easily manipulating them (see figure 2).

Once constructed, only the value of the implicit function representing the irregular domain and its derivatives need to be evaluated in order to compute the necessary moments. This allows the embedded boundary method to perform efficiently even on workstations.

If good estimates of the bounds of the local Taylor series expansion of the implicit functions are available, these can be used to dramatically improve performance by subdividing space recursively and only evaluating the implicit function where an irregular boundary could exist, i.e., regions where $\phi(\mathbf{x})$ could be zero. Thus, the amount of work done is proportional to the number of cells containing the irregular boundary and not the entire space. This also allows cells with under resolved geometry to be detected and corrected by adaptively refining these cells. This has been found to occur routinely when the higher-dimension irregular domains are restricted to lower dimensions; for example, slices of a well-resolved unit sphere can be circles with arbitrarily small radii. Once adequate refinement has been done, either the computation can use the refined computational cells, or they can be coarsened to provide the moments at the original resolution.

4. Conclusions and future work

It has been shown that finite volume methods for PDE can be done in arbitrary dimension to arbitrary accuracy in the presence of irregular boundaries by computing moments, integrals of monomials, over various regions. If the irregular boundary is defined by implicit functions, computing these moments can be computed using the divergence theorem, Taylor expansions, least squares, recursion, and 1D root finding. As a result, an explicit representation of the irregular domain and boundary is never needed nor computed. The resulting computations are robust and efficient even for complex domains which can be easily defined using implicit functions and constructive solid geometry.

These methods can be extended to mapped grids by incorporating coordinate mappings in the flux and moment computations. In addition, these methods can be used in any computational context where the algorithm can be reduced to the computation of moments over irregular domains or boundaries, for example, the integration of functions over these regions.

Acknowledgment

This research was supported by the Office of Advanced Scientific Computing Research of the US Department of Energy under contract number DE-AC02-05CH11231.

References

- [1] Colella P 2001 Volume-of-fluid methods for partial differential equations *Godunov Methods: Theory and Applications* E F Toro, 161–177
- [2] Aftosmis M, Berger M J and Melton J 1998 Robust and efficient Cartesian mesh generation for component-based geometry *AIAA J.* **36**(6) 952–960
- [3] Deschamps T, Schwartz P, Trebotich D, Colella P, Malladi R and Saloner D 2004 ‘Vessel segmentation and blood flow simulation using level sets and embedded boundary methods (Elsevier International Congress Series, 1268) 75–80
- [4] Schwartz P, Adalsteinsson D, Colella P, Arkin, A P and Onsum M 2005 Numerical computation of diffusion on a surface *Proc. Nat. Acad. Sci.* **102** 11151–11156
- [5] Malladi R, Sethian J A and Vemuri B C, Shape modeling with front propagation: A level set approach *IEEE Trans. Pattern Anal. Machine Intell.* **17**:158–175.